A Mitigation Model for DDoS Attack in Wireless Sensor Networks

Shruti Gond
Aishwarya Nath

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

A Mitigation Model for DDoS Attack in Wireless Sensor Networks

Thesis submitted in partial fulfilment
of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Shruti Gond
(Roll: 111CS0125)

Aishwarya Nath
(Roll: 111CS0464)

with the supervision of

Prof. Manmath Narayan Sahoo

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

May 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May, 2015

Certificate

This is to certify that the work in the thesis entitled ***A Mitigation Model for DDoS Attack in Wireless Sensor Networks*** by ***Shruti Gond*** and ***Aishwarya Nath*** is a record of an original work carried out with my supervision and guidance in partial fulfilment of the requirements for the degree of Bachelor in Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree elsewhere.

Manmath Narayan Sahoo
Assistant Professor
Department of CSE, NIT Rourkela

Acknowledgement

Our first thanks are to the Almighty, without whose blessings we would not have been writing this acknowledgment.

We offer our sincerest gratitude to our guide Dr. Manmath Narayan Sahoo for his stimulating support. He has always given us opportunity to learn more and more, supported us throughout our thesis with constant encouragement, being patient. At every point of time he helped us to do better and guided us with each important and even small things. We are highly indebted to him for taking out his precious time for us and guiding us through a proper channel. It would not have been possible this way without him.

We would also like to thanks to the faculty members of Computer Science and Engineering Department for imparting us with invaluable knowledge and adding a lot of value-oriented growth to our career.

Above all, we are blessed with such caring parents who have given us full freedom to do whatever we wish and have always supported us in our decisions. We extend our deepest gratitude to our parents and family members for their invaluable love, affection, encouragement and support.

Last but not the least, we would like to thank our friends and classmates who have been the most consistent source of motivation and affection and who have in one or another way have helped us in successful completion of this project and thesis.

Shruti Gond

Aishwarya Nath

Abstract

A Denial-of-Service is an attack in which the attackers send certain messages to the target systems or target servers with a purpose and intention of shutting down those system or servers. Those messages cause such an impact to the victim that it makes its services unavailable or not responding for the users. When a DoS attack is implemented in large number, then it is referred to as a DDoS or Distributed Denial-of-Service attack. In this, the attackers uses a large number of controlled bots called zombies and reflectors which are the innocent computers exploited to generate the attack. There are various kinds of DDoS attacks which depletes network bandwidth as well as its resources. We have particularly focused upon flooding kind of attacks. In this server's capacity is exploited by sending huge number of unwanted requests with a purpose of failure of server's processing efficiency. Since there is a limit to number of packet requests a server can effectively process. If that limit is exceeded, servers performance gets degraded.

In this thesis, we have followed an approach for mitigating DoS/DDoS attack based on the Rate Limiting algorithm, used to mitigate flooding resulting to the attack applied at the server-side. Packet filtering has been done on the basis of legitimate TTL values of the incoming packets followed by the ordering of packets to be sent to the server. Ordering of packets is performed with two approaches, one with the existing FCFS approach and other Priority queue approach and the server performance is compared. The implementation is carried out on the simulation tool MATLAB. The results show that there is considerable decrease in the two host and network based performance metrics that are Packet drop and Response time under DoS and DDoS attacks. When only legitimate packets are passed to the server after packet filtering, response time and throughput improves and after packet scheduling it even gets better.

Keywords: DoS; DDoS; flooding; packet drop; rate-limiting; response time; TTL value; scheduling.

Contents

1 Introduction **1**

1.1 Introduction 2

1.2 DDoS Impact 3

1.3 Problems Addressed 5

1.4 Design Approach 5

2 Distributed Denial-of-Service Attack and Related work **7**

2.1 Denial-of-Service Attack (DoS) 8

2.2 Distributed Denial-of-Service Attack (DDoS) 8

2.3 DDoS Taxonomy 9

2.3.1 Attacks Depleting the Bandwidth 9

2.3.2 Attacks Depleting the Resource 11

2.4 Taxonomy of DDoS Countermeasures 12

2.4.1 Mitigation at the Source 12

2.4.2 Mitigation at the Destination 12

2.4.3 Mitigation at the Network 13

2.4.4 Hybrid Mitigation Approach 13

2.5 Literature Review 14

2.6 Motivation and Problem Statement 15

3 Proposed Mitigation Model **18**

3.1 Terms Used 19

3.2 Applied Existing Algorithms 20

3.2.1 Idea of Leaky Bucket Algorithm 21

3.2.2 Functions and Variables Used 21

3.2.3 Probabilistic Method for Packet filtering 22
3.2.4 HCF METHOD 22
3.3 Proposed Mitigation Model 23

4 Simulations and Results **26**

4.1 Results and Analysis for Leaky bucket for congestion control 27
4.2 Results and Analysis for Packet filtering using Probabilistic Method and Hop Count Filtering 29
4.3 Results and Analysis for flooding scenario 31
4.4 Results and Analysis for Packets filtering using TTL value and packet scheduling 32

5 Conclusion and Future Works **35**

5.1 Conclusion 36
5.2 Future Works 36

List of Figures

1.1 Mechanism of DDoS Attacks 3

2.1 Classification of DDoS Attacks 9

3.1 Leaky Bucket Diagram 21

3.2 Flow chart of proposed method 23

4.1 Plot between Response time and Number of Nodes 28

4.2 Plot between Packets Dropped and Number of Nodes 29

4.3 Plot between Bits Dropped and Number of Nodes 30

4.4 Plot between number of nodes and packets generated by them 31

4.5 Plot between number of nodes Vs Response Time and Throughput under HCF method 32

4.6 Plot between packet drop Vs number of packets 32

4.7 Plot between throughput Vs number of packets 33

4.8 Plot between number of nodes Vs Response Time and Throughput under scheduling 34

List of Tables

2.1 Comparison between different defense mechanism 14
2.2 Comparison between different defense mechanism (continued...) 15
2.3 Comparison between different defense mechanism (continued....) 16

Chapter 1

Introduction

- Introduction
- DDoS Impact
- Problems Addressed
- Design Approach

1.1 Introduction

Before PCs and computers came into use extremely important information and data were put away in files. These records were physical properties and thus could be subsequently be stolen. To keep the documents safe and keep them from falling in wrong hands documents were kept in safes and locked. The individuals who could get to it were the individuals who had the key of the lock [5].

Anyway, now most critical information are put away in computers. But the fundamental prerequisite of giving security to information has not changed. The three primary highlights of security are -

- Integrity
- Confidentiality
- Accessibility

We realize that information is a benefit and ought to dependably be in a reliable state. We know that data is an asset and should always be in a consistent state. *Consistency* and *integrity* means that the data should be altered only by competent authorized person. Information without integrity has no value.

Confidentiality alludes to the way that the information ought to just be available to trusted parties. Private data about a system, individual or an organization ought to be furnished with high level of security. Example: The bank balance Information is also an asset and like all other assets we need to provide it security. Confidentiality alludes to the reality or secret key of one individual ought not be open to someone else.

Accessibility alludes to the reality when asked for the data must be accessible to the users. The data not being available or accessible at the obliged time is called refusal or *Denial-of-Service Attack*. In the present period of innovation, data has ended up being electronic What's more, it is put away in PCs and new age devices. Thus network security is of utmost importance. Network security relies on a lot of components for its performance. All these components must work together to provide security to the system.

DDoS attack is an attack of accessibility of assets not withstanding when they exist. *DDoS flooding attacks* are a big cause of worry for the security professionals as it keeps striking at any time. These are attempts by the attacker to prevent the real users from using the network resources by flooding the packet with illegitimate packets in large number. Attackers gain access to a large number of computers by taking advantage of loopholes and vulnerabilities existing in the system.

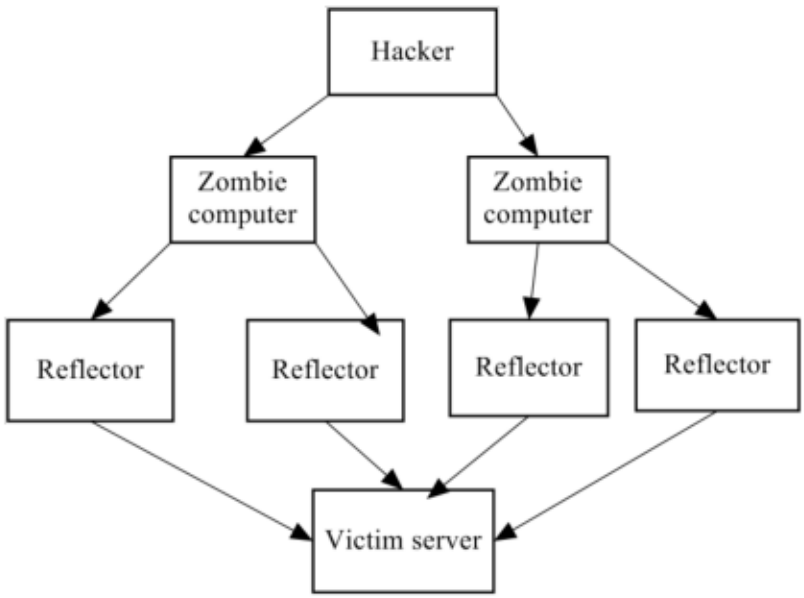

Figure 1.1: Mechanism of DDoS Attacks

Attackers assemble systems of affected PCs, known as 'botnets', by spreading vindictive programming through messages, sites and online networking. Once affected, these machines can be controlled remotely, without their origin information, and utilized like an armed force to dispatch an attack against any target. Some botnets are millions of machines strong.

1.2 DDoS Impact

Numerous DDoS flooding attacks have been dispatched against distinctive associations since the late spring months of 1999. The vast majority of the flooding attacks have made vic-

timized person's administrations inaccessible prompting income misfortunes and other budgetary problems. Let us take a couple of examples. In February 2000, Yahoo! encountered a DDoS attack that made it occupied to its clients for around 2 hours bringing about a critical misfortune in publicizing revenue. And in October 2002, 9 of the 13 Domain Name System (DNS) around the globe close down for 1 hour in light of flooding attacks. These attacks were dispatched utilizing virus. The infection was coded such that it instructed a large number of contaminated PCs to get to the site in the meantime. Recent advances in DDoS attacks have put an end to the time [18].

In a new research survey done by *VeriSign*, it was found that 75% of the questioned people had experienced at least one attack between 2008 and 2009. Therefore, protecting our network resources is of utmost importance today. The network security professionals must come together to find a solution.

It is very tedious to trace back to origin of the DDoS attack as thousands of compromised machines are involved in attacks. Thousands of computers are involved in the attack and hence, it impossible to track each computer. The different computers could be geographically located anywhere in the world. Furthermore, the attacker spoofs the IP addresses thus it becomes impossible to track the actual system. *IP spoofing* is the technique of creation of IP packets with the IP of the sender concealed behind some false address. In some cases these false IP packets try to impersonate other computer system. It is a technique in which the packets are coming from the system of the attacker but their IP addresses have been faked. So it appears like these packets are from some other computer system and these keep moving in the network consuming the resources. To engage in IP spoofing, a hacker needs to use a variety of techniques to impersonate other systems to gain access into them.

New switches routers and firewall arrangements coming into the market can offer protection against IP spoofing. Firewall is used as a PC framework which is intended for keeping unauthorized access to or from a private system. Firewalls are eligible for being implemented in both equipment and programming, or a mixture of two as well. They are frequently used for keeping away unauthorized Internet users from getting to private system that are associated with the Internet, mainly intranets. All messages passes a boundary of firewall whether goes in or out of internet, which checks on every message and obstructs those who do not

meet the safe criteria. Hence it gives security to our system by sifting malevolent packets.

A firewall is a security arrangement that controls the traffic coming into the system as well as traffic going out of the system based on some applied rules and principles. A firewall acts like a protective wall or barrier to provide security to our system. Firewalls could be implemented using hardware as well as software components. Many personal PC's and computers have built in software firewalls that fight bad traffic to some extent. Hardware based firewalls also provide many additional functionality to provide network security. Software firewall is implemented by a software running in our system. A software firewall provides security against most known attacks.

1.3 Problems Addressed

From the above examination we see that DDoS attacks are growing at a quick uncommon speed and are getting to be very frequent. The strategy for the assailant is to send malicious packets in substantial numbers to upset a real a client's network by debilitating bandwidth, router transforming limit or system assets. The target framework either reacts gradually so it is unusable and in more terrible cases it crashes down completely. Thus, mitigation of *Distributed Denial-of-Service* attacks is very essential to improve network performance and security. Different algorithms and methodologies have been proposed. But to be fruitful we need to have a thorough and complete methodology. Implementation of a mitigation or controlling technique for *Denial-of-Service* attack so that system is always able to give an optimal performance is our aim at present.

1.4 Design Approach

DDoS attacks occur when the amount of incoming packets is more than the processing capacity of the server. The algorithm used in our approach is the *Rate Limiting leaky Bucket Algorithm.* In this approach, we control the number of packets that reach the server and discarding the extra packets. As a result, the number of packets actually reaching the server is in control and can be processed efficiently by the server. The rate limiting leaky bucket algorithm helps us to reduce the *response time*, the *number of packets dropped* and

the number of bits dropped thereby improving the network performance [3].

But, only congestion control does not solve the problem of legitimacy. So, packet filtering has been done on the basis of TTL values of packets and then packet scheduling is done to improve server processing ability. We have measured server's performance with 2 parameters - *throughput* and *response time*.

When the packet drop, response time decreases and throughput increases, we ensure that the traffic is in control and hence the server system is not overwhelmed by too many packets and the server is processing legitimate requests.

Chapter 2

Distributed Denial-of-Service Attack and Related work

- Denial-of-Service (DoS) Attack
- Distributed Denial-of-Service (DDoS) Attack
- DDoS Taxonomy
- DDoS Counter Measures Taxonomy
- Literature Review

2.1 Denial-of-Service Attack (DoS)

Denial-of-Service Attacks are offered endeavors to prevent real clients from getting to a particular system resources. These attacks bargain the accessibility of assets regardless of the fact that they are accessible. A DoS assault is said to have happened just when reach or accessibility to a PC or system asset is deliberately suspended or alternately degraded as an after effect of pernicious move made by another client.

DoS attacks can be classified as:-

- **Information Flooding**: Here the attacker squanders system assets by sending huge amount of packets to the victim. The exploited system performance falls down as it can respond to fixed volume of traffic at a time. The victim server is not able to react to any further demands or requests.

- **Network Device Level**: These DoS attacks are accomplished by taking points of interest of the bugs existing in the product.

- **OS Level**: Here the attacker exploits the way the OS actualizes different sorts of protocol.

- **Application-based Attacks**: A large number of fake and malicious packets attempt to slow down a server by consuming its resources and bandwidth. This is done by taking advantage and exploiting the loopholes and bugs existing in the system and the applications that are running on it. The attacker takes advantage of the fact that the source IP address can be forged and hence, it is difficult to trace back its origin.

- **Attacks based on Protocol Feature**: Because of the fact that source addresses can be spoofed, these attacks take an advantage.

2.2 Distributed Denial-of-Service Attack (DDoS)

DDoS attacks are unequivocal attempts to disturb real clients attempts to access to resources. Attackers typically get entrance to an expansive amount of PCs by misusing their

weakness for setting up an armed force of botnets. Once, an assault armed force of botnets has been setup an attacker organize a composed huge attack against one or more targets.

A DDoS attacks have four participants :

(i) The genuine attacker.

(ii) The innocent or compromised hosts, who have the ability of governing many computers under them.

(iii) The daemon computers or zombies, who are in charge of generating a flood of packets toward the planned victimized system.

(iv) An exploited or target host.

2.3 DDoS Taxonomy

DDoS attacks classification shown below:

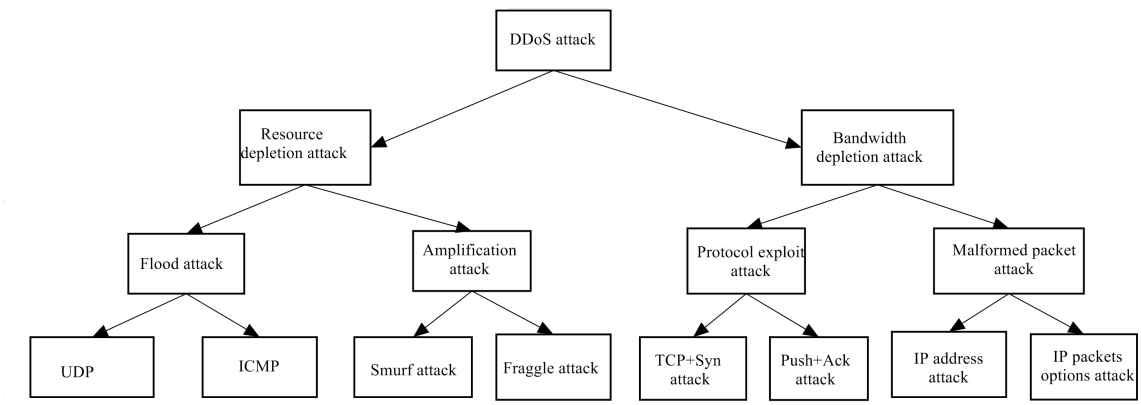

Figure 2.1: Classification of DDoS Attacks

2.3.1 Attacks Depleting the Bandwidth

There are two types of DDoS bandwidth consumption attacks [5]. They are:

(i) Flood Attacks

In case of DDoS flood attack, reflectors or zombies surge the exploited system frameworks with spoof IP activity such that the victimized system fails to react to any further requests. The huge volume of packets sent by these zombies saturates the processing capacity of the server such that the victim server either slows down so much that it is either not useful or completely crashes down. Examples-

- **UDP flood attack** : *User Datagram Protocol (UDP)* is a TCP/IP connection less protocol. When packets are transmitted between two hosts there is no need to have a handshake between them to initiate communication between them. When large number of packets are sent between them, the system's bandwidth starts getting depleted and it starts to slow down. In UDP flooding attacks the IP packets are sent to both known as well as unknown ports. The main aim is to attack as many number of arbitrary ports as possible. Most of the times the IP addresses are faked such that it is impractical to trace back the origin of the packets.

- **ICMP Flood Attacks** :*Internet Control Message Protocol (ICMP)* packets are used to calculate the number of hops traveled and to find out the time taken to complete full trip from the sender to the receiver. These packets can also be maliciously used by attacker to create a flooding scenario in the network. Example: "ping" signal lets the sender send a signal to the receiver and get back an acknowledgement as well the time taken to complete the full trips. Using botnets we can ping our server host so much that we crash down its performance completely.

(ii) Intensification or Amplification Attacks

A DDoS amplification attack done by utilizing the broadcast IP addresses. This permits the sending attack framework to utilize telecast IP addresses as destination addresses. As an outcome substantial number of PCs are influenced. In this sort of DDoS assault, the attacker sends the telecast message specifically, or the it also can utilize the agents to send the broadcast message to build large volume of bad traffic in the network. This attack furnishes the attacker with the capacity to utilize the systems

inside the telecast or broadcast range as zombies without expecting to invade them. Examples are:

- **Smurfing Attacks**: A DDoS smurf attack is an exploitation of existing internet protocols used to create Denial-of-Service attack. The attacker takes advantage of program called smurf to make the network inoperable. The exploits of smurfing attack takes advantage of loopholes in IP packet format as well as ICMP(Internet control message protocol). The ICMP is used by administration to exchange information about the state of network.

- **Fraggling Attacks**: It involves the attacker sending a huge amount of malicious traffic to the switch or routers broadcast address. Its principle of working is very much similar to a smurf attack but it has more devastating impact as compared to smurf attack. The main difference between them is smurf uses ICMP packets and fraggle uses UDP packets.

2.3.2 Attacks Depleting the Resource

DDoS resource depletion attacks include the aggressor sending packets that misuses system protocols or sending distorted packets that tie up system assets such that no assets are left for genuine clients. Examples:

- **Malformed Packet Attacks**: Here IP packets are tampered in such way such that its very difficult to find out its origin. The end aim of this attack is to make the victim system fail down completely. Most of the time IP packet contains same source and destination address such that it confuses the victim system. This can make uncertainty in the working arrangement of the victimized system framework and hence the victim system framework falls flat. In the event that this attack is reproduced sufficiently utilizing operators, it can close down the handling capacity of the exploited system framework.

- **TCP SYN Attacks** : The TCP protocol needs full handshake between the sender and receiver before the actual communication starts and transfer of data takes place.

The sender sends a *SYN* signal and when the receiver receives the packet it sends back an *ACK* signal. In case of an attack scenario, so many SYN signals comes from the compromised hosts such that the victim system in an attempt to send the ACK signals slows and very rapidly consumes its processor resources and then crashes down.

2.4 Taxonomy of DDoS Countermeasures

There are various recommendations and suggested arrangements accessible presently for mitigating the destructive impacts of a DDoS assault. A number of such arrangements and studies help in reducing DDoS attacks. In any case, there are no complete and comprehensive answers to ensure against every known type of DDoS attacks. More research and exertion is expected to give more viable and capable countermeasures and arrangements [10].

The accompanying data are some of the counter measures we can use to relieve DDOS attacks.

2.4.1 Mitigation at the Source

Here detection and reaction are done at the source hosts. The preference of this strategy is it distinguishes and reacts to the assault activity at the source before it squanders a lot of system assets. Anyhow, we know sources are distributed broadly among diverse places and therefore, it is troublesome for every sources to recognize and channel attack traffic accurately. And it is troublesome for separate detection at the source as the volume of activity is not vast enough. And we have low inspiration for arrangement as it is not clear who might bear the costs.

2.4.2 Mitigation at the Destination

In this, location and responses are studied at the destination that is, at the victim system side. It is less demanding and less expensive than different attempts in distinguishing DDoS attacks in light of their entrance to the total activity close to the destination has. But, before it reaches the victim they can't precisely recognize and react to the attack and hence wastes resources on the path.

2.4.3 Mitigation at the Network

Here we utilize switches and routers to deploy and detect attacks and mitigate at the intermediate attack. It intends to react to attack at close to source as possible. The principle weakness of this system is high overhead and high storage requirement at the routers. And attack recognition is troublesome on account of the absence of adequate attack traffic bound for the victim system.

2.4.4 Hybrid Mitigation Approach

Detection and responses are conveyed at different locations, detection typically happens at destinations and transitional systems and responses more often than not happens at the sources and upstream switches and routers close to the sources. There is a collaboration among different defense components. It is more robust against DDoS assaults. Yet, here complexity and overhead is more due to collaboration among dispersed component scattered everywhere throughout the web [6].

2.5 Literature Review

Table 2.1: Comparison between different defense mechanism

DEFENSE MECHANISM	AUTHOR AND YEAR	KEY POINTS	PROBLEMS
GI Time frequency algorithm	K. Kuppusamy , S. Malathi, 2011 [9]	-Uses technique to recognise the attacker and block them from using site. -For detection of attackers, a log history is maintained who ever requests the server.	Algorithm depends of the threshold value N which is the frequency of requesting the server.
Agent based preventive measures	A. Singh, D. Juneja, 2010 [6]	-Uses filter agent,timer agent and victim computer agent. -History buffer is main element on Host side to check the validity of received packet's address.	Suspicious IP address blocked only for specific time.
Packet Dropping based on software agent	B. Patel, M. Vishwakarma, 2013 [15]	-Calculates the load of the packet and checks that it lies between threshold limits. -Also checks the users history profile.	Legitimate packets may lie outside threshold limits.
Adaptive Selective Verification	S. Khanna, S.S. Venkatesh , 2012 [20]	-Clients can respond to attack by using limitation on bandwidth and issuing time out windows.	We assume the network is not lossy which is not practically true.

Table 2.2: Comparison between different defense mechanism (continued...)

DEFENSE MECHANISM	AUTHOR AND YEAR	KEY POINTS	PROBLEMS
Cooperative Mechanism using rate limiting.	H. Beitollahi, G. Deconinck, 2011 [3]	- At victim server's IP address, leaky bucket is installed. -Traffic rate is limited to a desired rate.	Problem in amending size of leaky bucket because it should be obtained from real time scenario,not possible in simulation
IP traceback ppm	S. Singh, A. Bhandari [6]	-Use IP trace back to capture actual attacker. -Router marks packet that helps to retrace the path.	Path reconstruction needs many computational cycles.
TBHF Filter	M. Ibrahim, K. Govind, 2012 [8]	-It prevents internet users from taking part in DDOS unknowingly. -Windows Filtering Platform is used to develop the algorithm.	Technique becomes inefficient when attacker spoofs actual IP address.
HCI-MPR	V. Chouhan, S.K. Peddoju, 2011 [12]	-Uses Poisson's distribution to calculate no. of malicious packet.	Uses probabilistic approach so legitimate packets might also get affected.

2.6 Motivation and Problem Statement

- **Motivation**

  To understand goals behind specific DDoS attacks and why they occur is normally difficult. The root cause are the machines and computers who perform the attack and are controlled by hidden exterior sources which makes it difficult to find the origin of attack. And when finding the host is already hard, it becomes even more difficult to find the reason behind the attack. DDoS attack is a severe threat to availability of network resources, preventing legitimate individuals from accessing a service. It either makes the server respond slow or makes the server completely fall down. There are malicious attempts made to make the resources unavailable to the users from server side and that is usually done by interrupting or halting the services temporarily of a host that is being connected to the internet.

Table 2.3: Comparison between different defense mechanism (continued....)

DEFENSE MECHANISM	AUTHOR AND YEAR	KEY POINTS	PROBLEMS
Adaptive Probabilistic Marking	H. Tian, B. Jun, X. Jiang , 2012 [14]	-For the initiation of proposed trace back scheme, TTL fields of packets are observed. -When DDoS attack occurs, attacking path is reconstructed.	Reconstruction of path take so many computational cycles hence creates overhead.
Traffic Pattern Analysis	T. Thapngam , 2012 [18]	-Using Pearson's correlation coefficient, patterns are analyzed and changes in traffic flow are observed for the calculation of observed parameter's standard deviation.	We use statistical approach so sometimes error may occur.
LOT Defense	Y.G. Herzberg , 2012 [21]	-With the help of proposed lightweight protocol, a tunnel is established between the two interacting gateways for the prevention of traffic against flooding and IP spoofing.	Multiple tunnels may be established on the route between two network and creates overhead.

The damage caused by DDoS attack scatters largely. According to various surveys, there have been many corporate as well as other sectors instances of DDoS attacks which shows how these attacks are affecting each and every sector of society with its evil intentions. Some underwent huge economical losses, some lost its authenticity since its users couldn't get proper response at right time. The cause may be personal rivalry, corporate competency or anything outcomes are same and section of society getting affected most is us, the consumers. So, these attacks needs to be mitigated.

- **Problem Statement**

  Calculate *packet drop* and *response time* of incoming node requests using Rate-Limiting by ***Leaky Bucket Algorithm*** .

  If the number of packets (N) > queue bucket size (L), then packets are dropped or else they are processed.

  Create a flooding scenario and study how network performance parameters degrade.

  Perform packet filtering to check the legitimacy of incoming packets on the basis of ***TTL values*** and scheduling the packets to be sent to the server. Observing performance of two network parameters *throughput* and *response time* in different scheduling scenarios as ***IBRL, FCFS, Priority Based***.

Chapter 3

Proposed Mitigation Model

- Terms Used
- Algorithm Applied
- Proposed Method

At present, internet that we are using is prone to attacks. The three main infrastructures which are availability, integrity and confidentiality are not yet achieved completely. The existing network infrastructure and its weakness is illegally exploited by attackers. Denial-of-Service attack is an active kind of attack that has impact on availability infrastructure of the internet. DoS attacks a system in various ways which we have already discussed. The DoS which is considered here creates a flood which uses bandwidth of the channel that was to be to be used by clients for legitimate work from server machine.

DDoS attack which uses millions of zombies. The packets sent by these zombies usually have a fake IP address and hence are difficult to be traced back. The available link bandwidth varies in accordance with the statistics of the input traffic. These statistics of arriving data traffic are not stationary but dynamic in nature.

The attackers use various network tools to simulate DDoS attacks. These attacks lead to wastage of precious network resources. Most of the approaches that are used to mitigate these attacks are not comprehensive in nature. The satisfactory efficiency to detect and filter out attack traffic is not being fully achieved by most of the current approaches. There are some loopholes that are misused by attackers to launch successful attacks.

The approach that is used here is the *Rate Limiting Leaky Bucket* algorithm to avoid flooding. Rate limiting assigns restriction to bandwidth for traffic like ICMP, UDP or specific connection types. Rate limiting leaky bucket algorithm helps us to control the flow of incoming packets into the server computer such that the server is not flooded with requests. But, this only helps in congestion control at this stage, so, to check legitimacy of an incoming packet we have performed packet filtering on the basis of TTL values of each incoming packet but, after there may be a possibility that large number of requests still prevails. So, for that proper scheduling of rest legitimate packets is required. That has been done using FCFS and priority queue approach.

3.1 Terms Used

For judging the performance of any network certain parameters are used. The parameters are used to evaluate the overall efficiency of the network. Some of the important parameters

are:

- **Throughput:** In communication networks, throughput refers to the average rate of successful processed requests over the total number of requests sent for processing. The path of transmission may be any medium. It is generally calculated in bits per second (bit/s or bps), many may be in data packets per second or per time slot.
- **Bandwidth:** Network bandwidth, is the bit-rate of available or used data communication resources that are generally expressed in bits/second or in multiples of it (bit/s, kbit/s, Mbit/s, etc). It refers to data carrying capacity of the network. More is the bandwidth, the more data the network can carry across its channels.
- **Packet Drop:** Packet drop occurs when requesting packets either fail to get processed or fails to reach server. There may be many reasons like in flooding all requests cannot be processed since they cross the servers capability to be processed and hence server has to drop exceeded packets or request may be illegitimate. Error in packet transmission may also lead to packet drop. When packet drop rate is high the performance of the network degrades.
- **Time-to-Live (TTL):** The time-to-live (TTL) is the number of hops traveled by a packet before it reaches to its destination or before it gets discarded by the router. This field is mostly used to control the maximum number of routers or hops that can be visited by the datagram [1].
- **Hop Count:** Before reaching at the destination, number of routers or intermediate devices a data packet visits is known as its hop-count [2].

3.2 Applied Existing Algorithms

The algorithm is applied at the server interface. The Leaky Bucket algorithm can be considered as a bucket that has an end at its bottom end from where water goes out uniformly and from the upper opening water is filled inside the bucket, so that when there is enough of water such that the bucket is filled, the excess water spills out. Similarly, here we have

taken the analogy of the bucket with a double ended queue which at one end sends uniform packet requests to the server and at the other end gets filled up and when it gets filled to its maximum capacity, extra requests are dropped out.

3.2.1 Idea of Leaky Bucket Algorithm

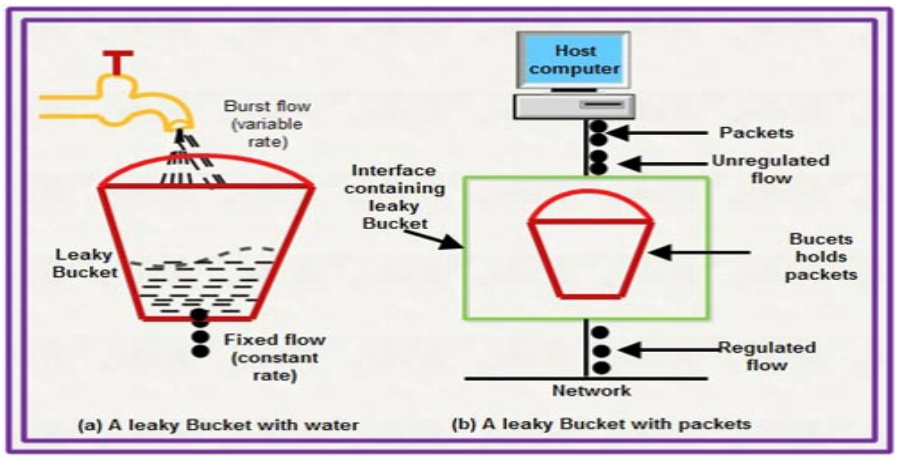

Figure 3.1: Leaky Bucket Diagram

The analogy of the algorithm is with a bucket having small hole at its bottom. No matter in what rate water is coming to the bucket it goes out with a fixed, constant and consistent rate. When buckets gets filled with water, rest water is dropped out and is lost.

- When the host send an incoming packet, the packet is sent into the bucket.
- The bucket's leak rate is constant. It means the bucket transforms bursty traffic into uniform traffic.
- Here, we have used a finite queue as bucket that has a constant rate output.
- If the number of packet coming in is more than the capacity of the bucket it is discarded.

3.2.2 Functions and Variables Used

- C = Capacity of the leaky bucket.
- Sum( )=Adds the total number of incoming packets.

- Add_to_bucket = Adds the packet to bucket if it is not full
- Process( )=The node is processed by the server.
- Discard( )=Packet is discarded if the bucket is full.
- Compute( )=Calculates the number of packets dropped, bits wasted and response time.

Algorithm 1 Leaky Bucket Algorithms

Require: Sum of packets from interfaces:S, Queue capacity:L

```
1: if Sum(S<L) then
2:    Add_to_bucket();
3:    Process();
4: else
5:    Discard();
6:    Compute();
7: end if
```

3.2.3 Probabilistic Method for Packet filtering

This is a probabilistic approach using Poisson's Distribution to get the number of malicious packets among all incoming packets. Suppose each packet arrive at server with ***p*** probability of being malicious and λ as Poisson's distribution [14].

Probability of '***n***' packets to be malicious is,

$$P = e^{-\lambda p} * (\frac{(\lambda p^n)}{n!}) * e^{-\lambda(1-p)}(\lambda(1-p)^m/m!)$$

After knowing the quantity of malicious packets, we can now apply ***hop count filtering method*** to discriminate legitimate packets from the illegitimate ones.

3.2.4 HCF METHOD

Time-to-live field of the IP header packet is used as its base in **Hop Count Filtering** method, since numbers of hops is the only field value of an IP header which cannot be falsified. Each incoming packet is checked for its TTL value and if its lies in the legitimate range (TTL value is the property of operating system and it differs for different operating systems), it is allowed to be processed by the server else it is discarded [2].

Algorithm 2 Hop Count Filtering Algorithm

Require: Poisson's Distribution: λ,
Probability of packet's maliciousness: p,
number of illegitimate packets: n,
number of non-malicious packets: m,
initial TTL value of packet: TTL_i,
final TTL value of packet: TTL_f

Ensure: P(n) = 1,
$TTL_i - TTL_f < 30$;

1: **if** ($TTL_f > TTL_i - 30$) || (30< TTL_f <=64) || (98 < TTL_f <= 128) || (225 < TTL_f <= 255) **then**
2: Process();
3: **else**
4: Discard();
5: Compute();
6: **end if**

This process saves the ***computation time*** as it does not need to check all the incoming packets but, only the number of packets found legitimate by the probabilistic method.

But, the drawback here is, since our computation is based on probability, it may not be correct all the time and those packet which we are not considering may contain some legitimate requests as well. So, though this method saves time but it doesn't give accuracy.

3.3 Proposed Mitigation Model

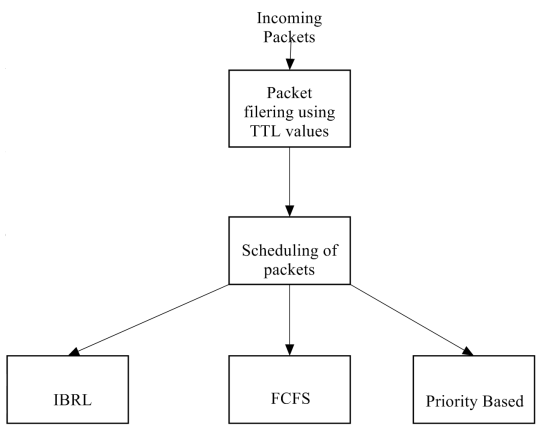

Figure 3.2: Flow chart of proposed method

In our method, we have followed the following procedure:

(i). Filtering of each incoming packet to the server using TTL value of the packets as the deciding parameter. If the final TTL value of packet is found to be in legitimate range, it is considered for further processing or else it is dropped.

(ii). After packet filtering, all the packets with which we are left, are the legitimate ones. But, it may so happen that, the packets are still in a very large number to be processed efficiently by the server. So, the packets need to be scheduled or we should decide the order in which packets should be reaching to the server so that the server can process all the requests.

(iii). For packet scheduling we have used three approaches:

- IBRL (Interface Based Rate limiting using leaky bucket):

  In this approach, packets are sent to the server at a uniform rate upto the definite server's capacity. Packets more than the server's capacity would not be processed [1].

- First Come First Serve scheduling(FCFS):

  Another way to sending packets may be on the basis of their arrival time. Whoever arrives first, will get a chance to be processed earlier. But, then there may be a possible situation the process which arrived first is a time taking process and because of that rest of the smaller time taking process have to wait a lot. So, this also didn't turn to be that efficient.

- Priority based scheduling:

  Every requesting packet in the system has been assigned some priority by the CPU, no matter what are the other kinds of requests are present, packets having higher CPU priorities will anyways be processed first. In our approach, we send the requesting packets on the basis of CPU priority first, if there arises a situation that, there is a collision between the CPU priorities of 2 or more packets, we then resolve it setting 1 more priority to the packets based on the packet's TTL value.

  Packets having TTL value more close to the legitimate range have been assigned higher priority. And this helps us to resolve the problem of CPU priority collision. Behavior

of two network parameters - throughput and response time was obtained and their performance in each scenario was compared.

Chapter 4

Simulations and Results

- Results and Analysis for Leaky bucket for congestion control.
- Results and Analysis for flooding scenario.
- Results and Analysis for Packet filtering using Probabilistic Method and Hop Count Filtering .
- Results and Analysis for Packets filtering using TTL value and packet scheduling.

The simulation is set up on a 64-bit operating system. The algorithm was simulated in MATLAB 7.8.0 (R2009a).

4.1 Results and Analysis for Leaky bucket for congestion control

(i) **From the Leaky bucket algorithm used for congestion control only, 3 parameters at the server side were analyzed :**

- Response time
- Packet drop number
- Number of bits dropped

(ii) **Observation is done against two scenarios:**

- Server node without leaky bucket implementation.
- Server node with leaky bucket implementation.

(iii) **Plot was obtained between following parameters:**

- Response time Vs number of nodes.
- Packets dropped Vs number of nodes.
- Bits dropped Vs number of nodes.

In *Figure 4.1*, a plot between Response time and Number of nodes is obtained. The plot shows comparison between the two scenarios i.e. with the application of leaky bucket and without its application. When leaky bucket is used, the response time decreases in comparison to the other scenario, which implies that now the server is able to process more number of requests with less delay. In *Figure 4.2*, a plot between Packets Dropped and Number of nodes is obtained. When the algorithm is not used, after a certain processing capacity, the server starts dropping the packets since it is not able to process those requests, but on

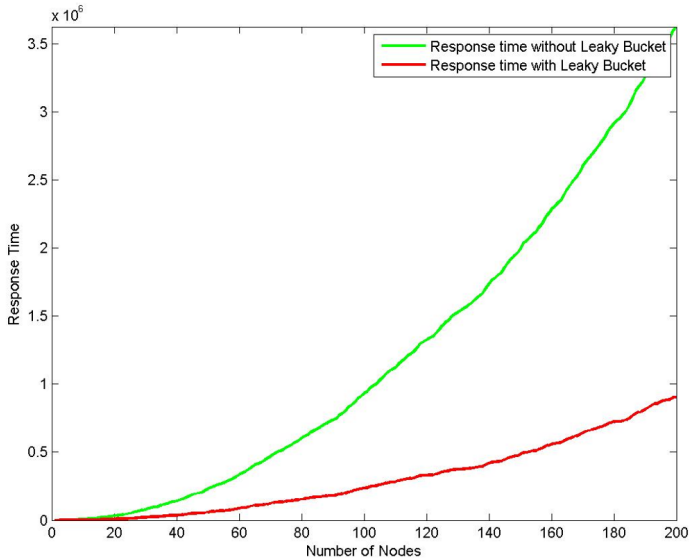

Figure 4.1: Plot between Response time and Number of Nodes

application of algorithm, number of packets drop also decreases. Similarly in case of *Figure 4.3*, a plot between Bits Dropped and Number of nodes is obtained. It also shows that when algorithm is applied the server is comparatively able to process more requests, congestion and bursty traffic is now controlled and server processes the requests uniformly.

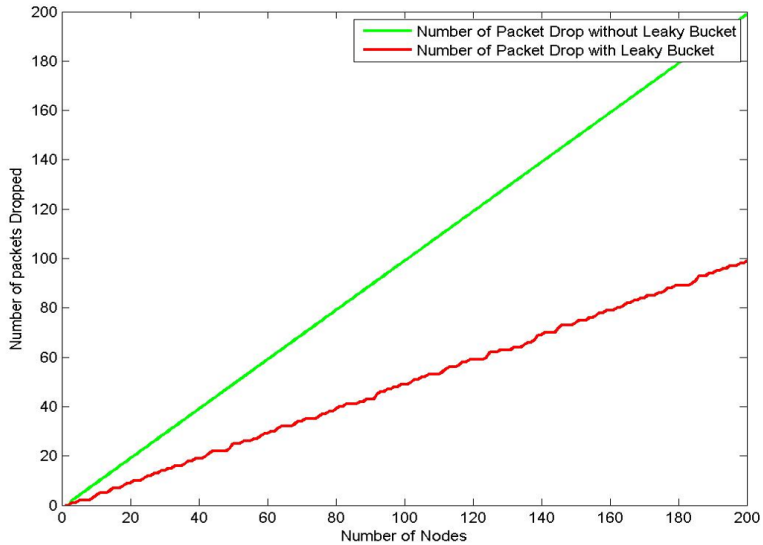

Figure 4.2: Plot between Packets Dropped and Number of Nodes

4.2 Results and Analysis for Packet filtering using Probabilistic Method and Hop Count Filtering

Probabilistic method was used to find out the probability of number of malicious packets and based on that count hop count method was applied for packet filtering. Two parameters were observed:

- Throughput
- Response Time

Figure 4.4 show the variation of generated packets in three different scenarios.

- First, shows the total number of packets generated by all nodes in the network.
- Second, shows decrease in number of packets after normal TTL filtering.
- Third, show the decrease in number of packets based on the probabilistic calculation.

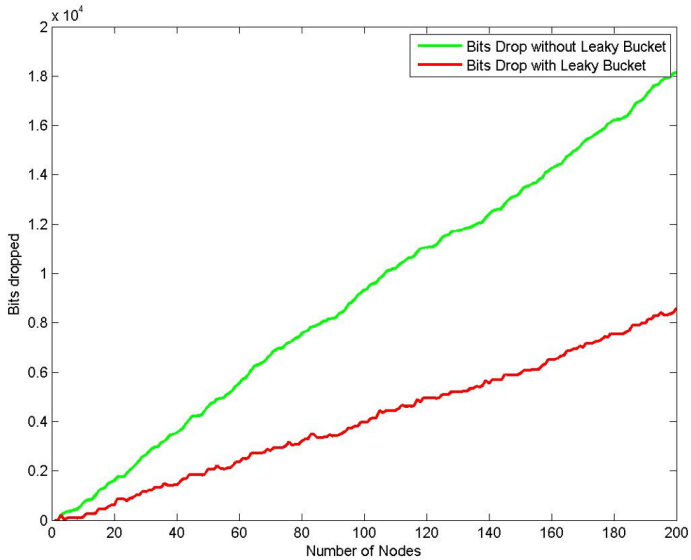

Figure 4.3: Plot between Bits Dropped and Number of Nodes

Figure 4.5 shows the plot between *Response Time, Throughput* and *Number of nodes*. Since, after the HCF method, only legitimate requests are allowed to go through the server and they are less in number, the processing capability of server increases and hence, there is an increase in throughput. Whereas, if we look at the plot of Response Time, it has decreased, because now more efficiently server is processing the packets.

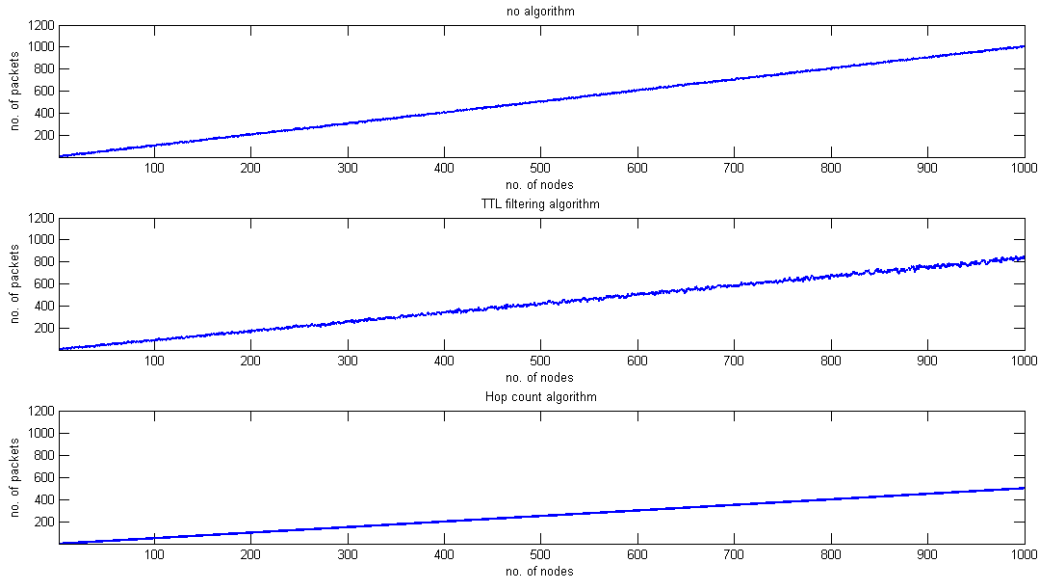

Figure 4.4: Plot between number of nodes and packets generated by them

4.3 Results and Analysis for flooding scenario

Figure 4.6 and *Figure 4.7* shows the performance of two parameters under flooding scenario which are *packet drop* and *throughput*. As more number of requests starts coming to the server, after a certain limit server becomes unable to process requests and the packet drop increases as further requests are being dropped. In other plot, throughput of the server (which the actually the number of requests processed by the server among the total requests heading towards the server) decreases as server's processing capability decreases under flooding, since huge server flood the server beyond its capacity, so it becomes unable to respond to all.

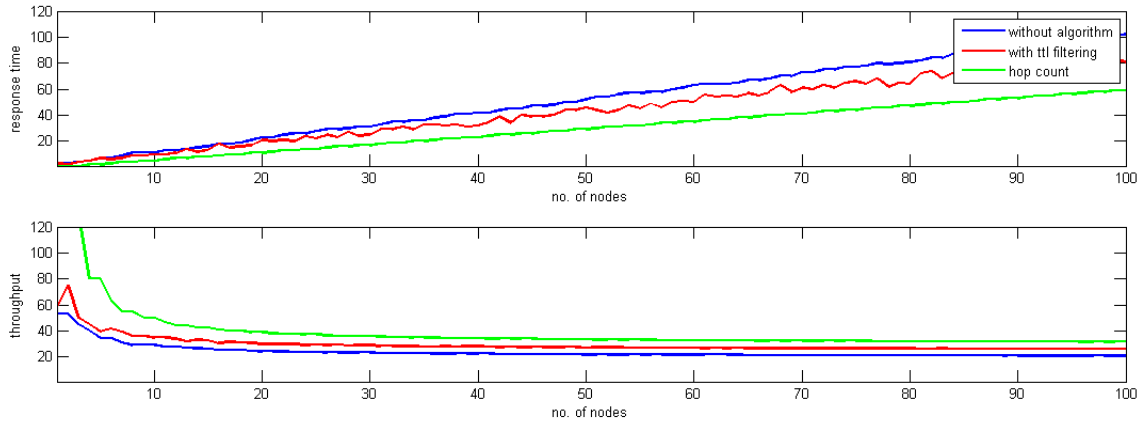

Figure 4.5: Plot between number of nodes Vs Response Time and Throughput under HCF method

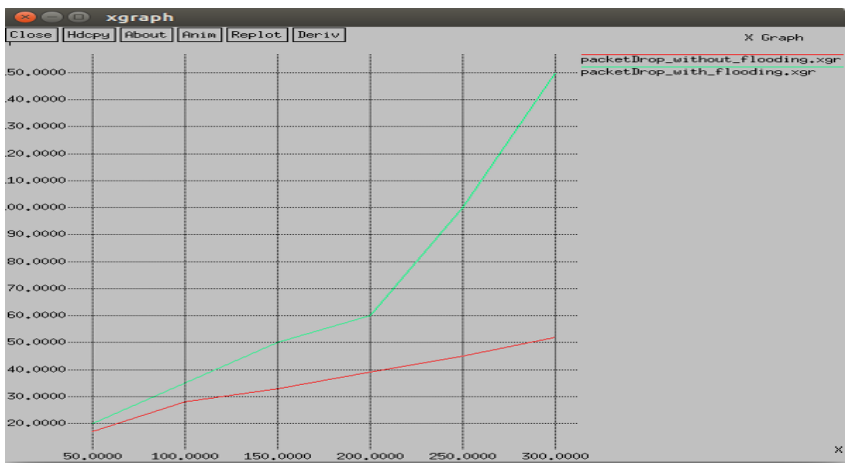

Figure 4.6: Plot between packet drop Vs number of packets

4.4 Results and Analysis for Packets filtering using TTL value and packet scheduling

Figure 4.8 shows the comparative plot of Response Time and Throughput Vs Number of nodes against the 3 scenarios: *IBRL*, *FCFS* and *Priority Based*. The plot clearly shows that the server performance is higher in case of Priority based scheduling. In FCFS approach, packets are sent on first come first serve basis but there may be a chance that a packet which is going first, may take longer time to be processed and hence in the priority approach packets are first sent on the basis of CPU priority and in case of priority collision, they are resolved by TTL value priority assigned to them, depending upon how close the TTL value is, higher

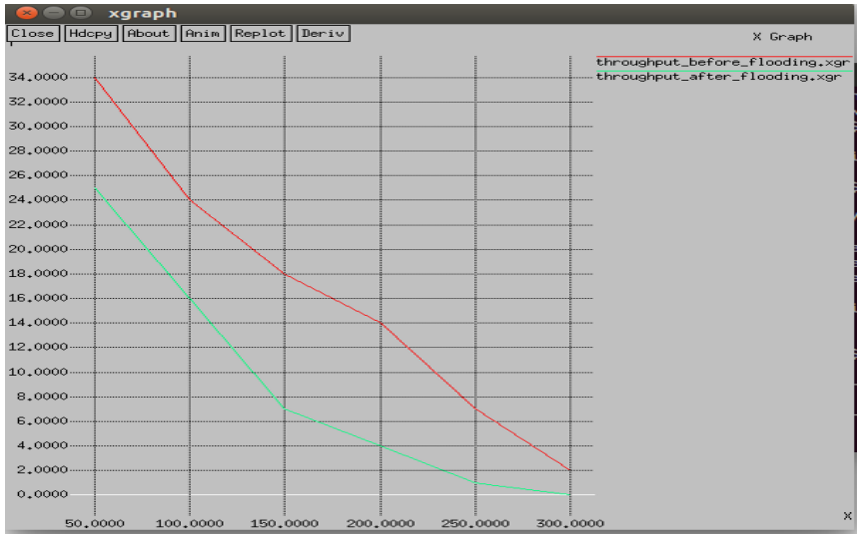

Figure 4.7: Plot between throughput Vs number of packets

priority is set.

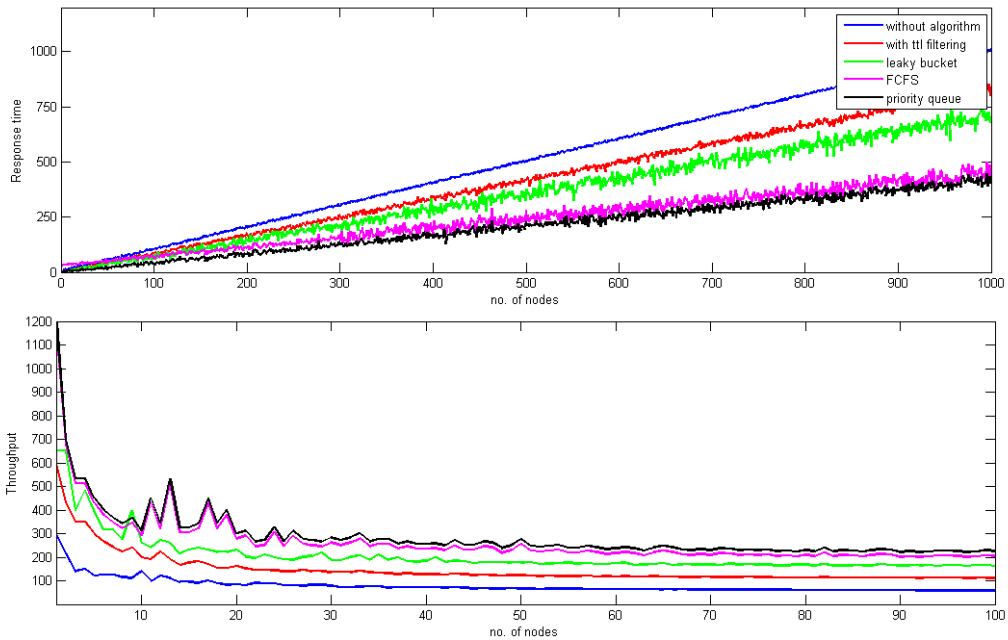

Figure 4.8: Plot between number of nodes Vs Response Time and Throughput under scheduling

Chapter 5

Conclusion and Future Works

- Conclusion
- Future Works

5.1 Conclusion

The applied scheme consists of application of leaky bucket at the server interface for normal congestion control. Apart from that it also does packet filtering and scheduling of packets. How server processes the request depends upon the capacity of server as well as the manner in which packets are coming.

In the leaky bucket plots, the comparison is shown between with the use of algorithm as well as without the use of algorithm. Without the algorithm, after processing a specific number of packet requests, the server limit congestion starts and processing is delayed and hence on application of algorithm, heavy flow of traffic is converted into a consistent traffic and processing smoothly goes on.

Legitimacy of a packet is checked on the basis of its TTL value hop count. Different algorithms have been applied to compare and check that under which scenario network performance is improved and efficient.

5.2 Future Works

Optimizing the algorithms used in such a way that it reduces computation time as well as increases efficiency. There are various network based parameters which can also be analyzed and could help in improvements in the procedure.

Bibliography

[1] B.S.K. Devi, G. Preetha, G. Selvaram and S. M. Shalinie ,"An impact analysis: real time DDoS attack detection and mitigation using Machine Learning",*International Conference on Recent Trends in Information Technology/IEEE*, 2014.

[2] B.R. Swain and B.D. Sahoo ,"Mitigating DDoS attack and saving computational time using a probabilistic approach and HCF method",*International Advance Computing Conference (IACC 2009)*, 6-7 March 2009.

[3] H. Beitollahi and G. Deconinck ,"A cooperative mechanism to defence against distributed denial of service attack",*International Joint conference of IEEE trustcom-11/IEEE ICESS-11/FCST-11*, November 2011.

[4] S. Singh and A. Bhandari ,"Review of PPM, a traceback technique for defending against DDoS attacks ",*International Journal of Engineering Trends and Technology (IJETT)* , Vol. 4, Issue 6, June 2013 .

[5] S.T. Zargar, J. Joshi and D. Tipper ,"A survey of defense mechanisms against Distributed Denial of Service (DDoS) flooding attacks",*IEEE Communications Surveys and Tutorials*, February 2013.

[6] A. Singh, D. Juneja ,"Agent based preventive measure for UDP flood attack in DDoS attacks ",*International Journal of Engineering Science and Technology* , Vol. 2(8), ISSN No. 3405-3411, 2010.

[7] S.S. Chowriwar, M.S. Mool, P.P. Sabale, S.S. Parpelli and N.Sambhe ,"Mitigating Denial-of-Service attacks using secure service overlay model",*International Journal of Engineering Trends and Technology (IJETT)*, Vol.8, No.9, Feb 2014 .

[8] A.K.M. Ibrahim, L. George, K. Govind and S. Selvakumar ,"Threshold based kernel level http filter (TBHF) for DDoS mitigation", *I. J. Computer Network and Information Security*, November 2012 .

[9] K. Kuppusamy and S. Malathi ,"An effective prevention of attacks using GI Time Frequency algorithn under DDoS", *International Journal of Network Security and Its Applications (IJNSA)*, Vol.3, No.6, November 2011.

[10] D. Kumar, C.V.G. Rao, M.K. Singh, Satyanarayana ,"A Survey on defense mechanisms countering DDoS attacks in the network", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol.2, Issue 7, July 2013.

[11] P.S. Mann and D. Kumar ,"Improving network performance and mitigate DDoS attacks using analytical approach under collaborative software as a service cloud computing environment ",*International Journal of Computer Science and Technology (IJCST)*, Vol-1, Iss-1, ISSN No. 2315-4209, 2012.

[12] V. Chouhan and S.K. Peddoju ,"Packet monitoring approach to prevent DDoS attack in cloud computing ",*International Journal of Computer Science and Electrical Engineering (IJCSEE)*, Vol.2, Issue 1, March 2011.

[13] Geetika, N. Kumari ,"Detection and Prevention Algorithms of DDOS Attack in MANETs ",*International Journal of Computer Science and Software Engineering* ,Vol.3, Issue 8, August 2013.

[14] H. Tian, J. Bi and X. Jiang ,"An adaptive probabilistic marking scheme for fast and secure traceback",*Networking Science Research Article Tsinghua University Press and Springer* , Vol.2, Issue 1-2, pp 42-51, May 2013.

[15] B. Patel and M. Vishwakarma ,"Impact of DDoS attack in online auction system and proposed lightweight solution based on software agent ",*International Journal of Computer Trends and Technology (IJCTT)*, Vol.5, No.6, November 2013.

[16] J. Luo, X. Yang, J. Wang, J.X.J. Sun and K. Long ,"Mathematical model for low-rate shrew DDoS ", *IEEE , Transactions on information forensics and security*, Vol.9, No.7, July 2014.

[17] B. Balasankaran and M.G. Mathan Kumar ,"Detection of spoofing using packet marking algorithm",*International Journal of computer trends and technology(IJCTT)*, Vol.10, No.5, April 2014.

[18] T. Thapngam and S. Kami ,"Distributed Denial of Services detection by traffic pattern analysis",*International Journal of Network Security and Its Applications (IJNSA)*, Vol.7, April 2014.

[19] K. Park and H. Lee ,"The effectiveness of router based packet filtering ",*International Journal of Engineering Trends and Technology (IJETT)*,Vol.31, No.4, pp 15-26, 2012.

[20] S. Khanna and S. Venkatesh ,"DDoS attack prevention by Adaptive Selective Verification",*International Journal of Engineering Trends and Technology (IJETT)*,Vol.4, Issue 6, June 2013.

[21] Y. Gilad and Herzberg ,"LOT Defence Mechanism for prevention of DDoS attacks ",*International Journal of Engineering Trends and Technology (IJETT)*,Vol.10, No.5, April 2012.

[22] M. Aamir, M. Arif ,"Study and performance evaluation on recent DDoS trends of attack and defense ",*International Journal of Information Technology and Computer Science*,pp 54-65, August 2013.

[23] M. Antikainen, T. Aura, and M. Srel ,"Denial-of-Service attacks in Bloom-Filter-Based forwarding ",*IEEE/ACM TRANSACTIONS ON NETWORKING*,Vol.22, No.5, October 2014.